

Claims

WHAT IS CLAIMED IS:

1. A computer system configured for providing themes for graphical components, the computer system having memory, the computer system comprising:
 - a selecting module receiving a user request for a selected theme having an associated non-binary theme file with theme properties;
 - 5 a converting module converting the associated non-binary theme file into a binary theme file to facilitate retrieval of theme properties; and
 - a loading module loading the binary theme file into the memory so that themes can be applied to the graphical components.
2. The system of claim 1 further comprising a plurality of processes, each process accessing the binary theme file.
3. The system of claim 1, further comprising:
 - an update handle module receiving a theme handle request from a graphical component and distributing a theme handle if the graphical component is found in the binary theme file so that the graphical component can use the theme properties of the binary theme file; and
 - 5 a close handle module closing the theme handle and decrementing a reference count on the shared memory in response to process termination so that a theme handle can be closed when a binary theme file is loaded.
4. The system of claim 1, further comprising:
 - a notification module notifying the processes that a new theme file has been loaded.
5. The system of claim 1, wherein the converting module comprises:
 - a schema file parsing module parsing a schema file containing a list of all themeable graphical components and properties;
 - a theme specification file parsing module parsing a theme specification file specifying graphical component sizes and colors; and
 - 5

a building module building a binary theme file containing the graphical components, properties, sizes, and colors in a binary format.

6. The system of claim 5 wherein the binary format is hierarchical, there being a data section for each hierarchy, the sections being a global section, a class section, a parts section, and a states section.
7. The system of claim 6 wherein the converting module further builds a packed data object section having all the theme properties for a class, part, and state.

Station	Time	Lat.	Long.	Alt.	Temp.	Wind	Clouds	Remarks
1	0800	34° 15' N	122° 00' W	10	58.0	10	0	Clear
2	0900	34° 30' N	121° 45' W	10	58.5	10	0	Clear
3	1000	34° 45' N	121° 30' W	10	59.0	10	0	Clear
4	1100	35° 00' N	121° 15' W	10	59.5	10	0	Clear
5	1200	35° 15' N	121° 00' W	10	60.0	10	0	Clear
6	1300	35° 30' N	120° 45' W	10	60.5	10	0	Clear
7	1400	35° 45' N	120° 30' W	10	61.0	10	0	Clear
8	1500	36° 00' N	120° 15' W	10	61.5	10	0	Clear
9	1600	36° 15' N	120° 00' W	10	62.0	10	0	Clear
10	1700	36° 30' N	119° 45' W	10	62.5	10	0	Clear
11	1800	36° 45' N	119° 30' W	10	63.0	10	0	Clear
12	1900	37° 00' N	119° 15' W	10	63.5	10	0	Clear
13	2000	37° 15' N	119° 00' W	10	64.0	10	0	Clear
14	2100	37° 30' N	118° 45' W	10	64.5	10	0	Clear
15	2200	37° 45' N	118° 30' W	10	65.0	10	0	Clear
16	2300	38° 00' N	118° 15' W	10	65.5	10	0	Clear
17	0000	38° 15' N	118° 00' W	10	66.0	10	0	Clear
18	0100	38° 30' N	117° 45' W	10	66.5	10	0	Clear
19	0200	38° 45' N	117° 30' W	10	67.0	10	0	Clear
20	0300	39° 00' N	117° 15' W	10	67.5	10	0	Clear
21	0400	39° 15' N	117° 00' W	10	68.0	10	0	Clear
22	0500	39° 30' N	116° 45' W	10	68.5	10	0	Clear
23	0600	39° 45' N	116° 30' W	10	69.0	10	0	Clear
24	0700	40° 00' N	116° 15' W	10	69.5	10	0	Clear

8. A method for creating a visual style for a set of graphical components for use on a computer system having a graphical operating environment and processes with shared memory, the method comprising:

5 selecting graphical components from a schema file of graphical components, that are desired to have a defined visual style, each component being defined by a unique class name;

assigning properties to the selected components according to the defined visual style so that each selected component has assigned properties;

10 grouping the pairs of selected graphical components and corresponding assigned properties for the defined visual style together in a class data file;

converting the class data file into a binary theme file having a class data section having class names and assigned properties in a binary format; and

loading the binary theme file into the shared memory so that a visual style can be used to render graphical components.

9. The method of claim 8, wherein the graphical components defined within the schema file of graphical components have one or more part names associated with at least one class name, and the converting act further comprises creating a part property data section in the binary theme file, the part property data section having the one or more part names and the assigned properties.

10. The method of claim 8, wherein the graphical components defined within the schema file of graphical components have one or more state names associated with at least one defined part name, and the converting act further comprises creating a state property data section in the binary theme file, the state property data section having the one or more state names and the assigned
5 properties.

11. The method of claim 8 further comprising:

identifying some properties as global properties; and

5 creating in the binary theme file a global properties section having the global properties to be used when a class name, part name, or state name cannot be found in the binary theme file.

12. The method of claim 11, wherein a list of available properties is within the first schema file of graphical components, that may be selected in the selecting step for each graphical component, part and state.

13. The method of claim 12, wherein the act of converting comprises:

identifying a derived property for a graphical component;

associating a unique numeric identifier with the derived property to create a derived property identifier;

5 identifying one or more primitive properties for each derived property, wherein each primitive property has associated property data having a length;

associating a unique numeric identifier with each primitive property, to create a primitive property identifier;

calculating the lengths of each of the associated property data;

10 selecting a derived property identifier;

writing a binary tagged data module to a tagged data memory offset in the class data section of the binary file wherein the binary tagged data module contains the selected derived property identifier, the one or more primitive property identifiers, the associated property values, and each of the property values' lengths; and

15 writing an associated parent part offset after each binary tagged data module, the associated parent part offset being a memory offset into the global class section.

14. The method of claim 13 wherein the act of converting further comprises:

obtaining the memory offset of a binary tagged data module for a state; and

writing the memory offset to a second memory offset in a state jump table in the binary theme file.

15. The method of claim 14 wherein the act of converting further comprises:

writing the second memory offset to a third memory offset in a part jump table in the binary theme file.

16. A method of applying a theme in a computer system having a graphical operating system and applications running thereon comprising:

receiving a user request to use a user selected theme defined by a text theme file;

creating a binary file from the text theme file;

5 loading the binary file into a shared memory area, capable of being shared by all applications;

notifying running applications that there is a new theme;

receiving a handle update request from a running application, the request having an application name;

10 searching for the application name in the binary file;

determining if the application name is in the binary file;

locating a theme handle in the binary file if the application name exists in the binary file; and

15 returning the located theme handle to the requesting application whereby theme properties can be applied to the application.

17. A method of retrieving graphical component theme property data on a computer system having a graphical operating system and processes, the method comprising:

receiving a rendering request from a graphical component, the request having a theme handle and a component state;

5 accessing a binary theme file to retrieve theme property data for the requesting process; and

retrieving graphical component theme property data from the binary theme file.

18. The method of claim 17 wherein the act of accessing a binary theme file comprises:

retrieving an offset into a class data section of the binary theme file, the classdata section having theme property data for a class in binary format;

performing a binary search for class property data at the offset;

5 determining if class property data exists at the offset;

jumping to a global data section of the binary theme file having global theme property data, if no class property data is found; and

retrieving global theme property data from the global data section.

19. The method of claim 17 wherein the act of accessing a binary theme file comprises:

retrieving an offset into a part jump table section of the binary theme file, the part jump table section having theme property data for a part in binary format;

performing a binary search for part property data at the offset;

5 determining if part property data exists at the offset;

jumping to a class data section of the binary theme file having theme property data for a class, if no part property data is found; and

retrieving class theme property data from the class data section.

20. The method of claim 17 wherein the step of accessing a binary theme file comprises:

retrieving a memory offset into a part jump table section of the binary theme file;

retrieving from the part jump table section a second memory offset into a state jump table section;

- 5 jumping to the second memory offset of the binary theme file having state theme property data; and
- retrieving state theme property data from the state theme property data section.

21. A computer program product readable by a computing system and encoding a computer program of instructions for executing a computer process for creating a visual style for a set of graphical components for use on a computer system having a graphical operating environment and processes with shared memory, said computer process comprising:

5 selecting graphical components from a schema file of graphical components, that are desired to have a defined visual style, each component being defined by a unique class name;

 assigning properties to the selected components according to the defined visual style so that each selected component has assigned properties;

10 grouping the pairs of selected graphical components and corresponding assigned properties for the defined visual style together in a class data file;

 converting the class data file into a binary theme file having a class data section having class names and assigned properties in a binary format; and

15 loading the binary theme file into the shared memory so that a visual style can be used to render graphical components.

22. The computer program product of claim 21 wherein the graphical components defined within the schema file of graphical components have one or more part names associated with at least one class name, and the converting act further comprises creating a part property data section in the binary theme file, the part property data section having the one or more part names
5 and the assigned properties.

23. The computer program product of claim 22 wherein the graphical components defined within the schema file of graphical components have one or more state names associated with at least one defined part name, and the converting act further comprises creating a state property data section in the binary theme file, the state property data section having the one or more state
5 names and the assigned properties.

24. The computer program product of claim 21 further comprising:
 identifying some properties as global properties; and
 creating in the binary theme file a global class section having the global properties to be used when a class name, part name, or state name cannot be found in the binary
5 theme file.

25. The computer program product of claim 24, wherein a list of available properties is within the first schema file of graphical components, that may be selected in the selecting step for each graphical component, part and state.

26. The computer program product of claim 22, wherein the act of converting comprises:
 identifying a derived property for a graphical component;
 associating a unique numeric identifier with the derived property to create a
 derived property identifier;

5 identifying one or more primitive properties for each derived property, wherein
 each primitive property has associated property data having a length;
 associating a unique numeric identifier with each primitive property, to create a
 primitive property identifier;
 calculating the lengths of each of the associated property data;
 10 selecting a derived property identifier;
 writing a binary tagged data module to a tagged data memory offset in the class
 data section of the binary file wherein the binary tagged data module contains the selected
 derived property identifier, the one or more primitive property identifiers, the associated
 property values, and each of the property values' lengths; and
 15 writing an associated parent part offset after each binary tagged data module, the
 associated parent part offset being a memory offset into the global class section.

27. The computer program product of claim 26, wherein the act of converting further
 comprises:

obtaining the memory offset of a binary tagged data module for a state; and
 writing the memory offset to a second memory offset in a state jump table in the
 5 binary theme file.

28. The computer program product of claim 27 wherein the act of converting further
 comprises writing the second memory offset to a third memory offset in a part jump table in the
 binary theme file.

29. A computer program product readable by a computing system and encoding a computer program of instructions for executing a computer process for applying a theme in a computer system having a graphical operating system and applications running thereon comprising:

- 5 receiving a user request to use a user selected theme defined by a text theme file;
- creating a binary file from the text theme file;
- loading the binary file into a shared memory area, capable of being shared by all applications;
- notifying running applications that there is a new theme;
- receiving a handle update request from a running application, the request having
- 10 an application name;
- searching for the application name in the binary file;
- determining if the application name is in the binary file;
- locating a theme handle in the binary file if the application name exists in the binary file; and
- 15 returning the located theme handle to the requesting application whereby theme properties can be applied to the application.

30. A computer program product readable by a computing system and encoding a computer program of instructions for executing a computer process for retrieving graphical component theme property data on a computer system having a graphical operating system and processes, the method comprising:

- 5 receiving a rendering request from a graphical component, the request having a theme handle and a component state;
- accessing a binary theme file to retrieve theme property data for the requesting process; and
- retrieving graphical component theme property data from the binary theme file.

31. The computer program product of claim 30 wherein the act of accessing a binary theme file comprises:

- retrieving an offset into a class data section of the binary theme file, the class data section having theme property data for a class in binary format;
- 5 performing a binary search for class property data at the offset;
- determining if class property data exists at the offset;
- jumping to a global data section of the binary theme file having global theme property data, if no class property data is found; and
- retrieving global theme property data from the global data section.

32. The computer program product of claim 30 wherein the act of accessing a binary theme file comprises:

- retrieving an offset into a part jump table section of the binary theme file, the part jump table section having theme property data for a part in binary format;
- 5 performing a binary search for part property data at the offset;
- determining if part property data exists at the offset;
- jumping to a class data section of the binary theme file having theme property data for a class, if no part property data is found; and
- retrieving class theme property data from the class data section.

33. The computer program product of claim 30 wherein the act of accessing a binary theme file comprises:

- retrieving a memory offset into a part jump table section of the binary theme file;
- retrieving from the part jump table section a second memory offset into a state
5 jump table section;
- jumping to the second memory offset of the binary theme file having state theme property data; and
- retrieving state theme property data from the state theme property data section.

34. A computer-readable medium having a data structure stored thereon with one or more theme data sections for providing a theme property for a graphical component in a themeable windows environment of a computer system, the graphical component being associated with a target class comprising:

- 5 a class data section having one or more first theme properties, each first theme property being associated with a class of a component;
- a global data section having one or more second theme properties, each second theme property being associated with any class of graphical component; and
- a jump-to-parent entry storing a reference into the global data section to provide
10 access to a second theme property if a first theme property for a target class does not exist in the class data section.

35. The computer-readable medium of claim 34 further comprising:

- a part data section having one or more third theme properties for a part, a part being a type of a class;
- a part jump table containing a reference to a list of one or more third theme
5 properties in the part data section; and
- a jump-to-parent entry storing a reference into the class data section to provide access to a first theme property if a third theme property for a target part does not exist in the part data section.

36. The computer-readable medium of claim 35 further comprising:

- a state data section having one or more fourth theme properties for a state, a state being a state of a part;
- a state jump table containing a reference to a list of one or more fourth theme
5 properties in the state data section; and
- a jump-to-parent entry storing a reference into the part data section to provide access to a third theme property if a fourth theme property for a target state does not exist in the state data section.

37. The computer-readable medium of claim 36 wherein the part jump table further comprises a reference into the state jump table.

38. The computer-readable medium of claim 36 further comprising a drawing objects section having one or more drawing objects, each drawing object having one or more theme properties associated with a target state of a target part of a target class.

39. The computer-readable medium of claim 38 wherein the part jump table further comprises a reference into the drawing objects section.

40. The computer-readable medium of claim 38 wherein the one or more theme properties are copied from the one or more first theme properties of the class data section, the one or more third theme properties of the part data section, and the one or more fourth theme properties of the state data section.

41. The computer-readable medium of claim 38 further comprising a custom region data section having custom region data, wherein the custom region data comprises points and associated grid references defining nontransparent parts of an image, the image being an imagefile property of a drawing object, the image comprising a plurality of grids, each grid
5 having a left boundary, a top boundary, a right boundary, and a bottom boundary.

42. The computer-readable medium of claim 41 wherein a point comprises a horizontal offset value and a vertical offset value, the horizontal offset value being an offset from the top boundary of the associated grid and the vertical offset value being an offset from the left boundary of the associated grid.

43. The computer-readable medium of claim 41 wherein the custom region data section further comprises an offset to a custom region data header, the header having a part identifier and state identifier.

44. The computer-readable medium of claim 41 wherein the drawing object further comprises a region list offset, the region list offset being a reference into the custom region data section.

45. The computer-readable medium of claim 39 further comprising a sampled colors section, having sampled color data for an image, the image being an imagefile property of a drawing object, the sampled color data providing sample colors to facilitate retrieval of the image.

46. The computer-readable medium of claim 45 wherein the drawing object further comprises a sampled colors offset into the sampled colors section.

47. The computer-readable medium of claim 39 further comprising a text object section having one or more text objects, each text object having one or more theme properties associated with a graphical text component.

48. The computer-readable medium of claim 39 further comprising a text object section having one or more text objects, each text object having one or more theme properties associated with a graphical text component.